DOCUMENT RESUME

ED 425 730 IR 019 218

AUTHOR	Snyder, Robin M.
TITLE	Using 4DOS Batch Files To Create an Infrastructure That Makes It Easy for Students To Create and Maintain HTML Web Pages.
PUB DATE	1998-00-00
NOTE	10p.; In: Association of Small Computer Users in Education: Proceedings of the ASCUE Summer Conference (31st, North Myrtle Beach, SC, June 7-11, 1998); see IR 019 201.
PUB TYPE	Reports - Descriptive (141) -- Speeches/Meeting Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	*Authoring Aids (Programming); Automation; College Students; *Computer Oriented Programs; *Computer Software Development; Higher Education; *Hypermedia; *Programming; Programming Languages; User Needs (Information); World Wide Web
IDENTIFIERS	*Batch Processing (Computer Science); Disk Operating Systems; HTML; Infrastructure; *Web Pages

ABSTRACT

This paper describes the use of simple 4DOS batch files to automate the creation and maintenance of an infrastructure to assist students in creating and maintaining HTML World Wide Web pages. Background is provided on Web pages, DOS, 4DOS, and batch files. The assumptions made in creating the infrastructure are summarized, and program statements and resulting output are presented. (AEF)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

Using 4DOS batch files to create an infrastructure that makes it easy for students to create and maintain HTML Web Pages

PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

C.P. Singer

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

Robin M. Snyder
Byrd School of Business
Shenandoah University
Winchester, VA 22601
rsnyder@su.edu

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☐ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

ED 425 730

ABSTRACT

Learning HTML is not that difficult. Neither is using an HTML editor. Many of the practical problems in having students create and maintain HTML web pages are a result of a lack of a suitable infrastructure. This paper (and presentation) will describe (and demonstrate) the use of simple 4DOS batch files to automate the creation and maintenance of such a system. Although 4DOS supports quite sophisticated batch processing, many common tasks involving multiple users, directories, and files can be done by taking the supplied batch file templates and making simple modifications to them. This paper (and presentation) would be of interest to anyone, including network administrators, who want to make it easy for students to create and maintain HTML web pages, or who wish to automate some of the many tasks that must be done for each and every user, directory, or file on the network.

WEB PAGES

Web pages are text files that are written in a text formatting language called HTML. It is assumed that the reader is somewhat familiar with HTML files. HTML files are accessed on the Internet via their URL (Uniform Resource Locator). The URL for Shenandoah University is

```
http://www.su.edu
```

which refers to the University's web server running Microsoft Windows NT. All clients run Windows 95 while the file servers run Novell NetWare. The author's web directory on the Internet is in the WEB directory on the KITHARA: volume of the NetWare file server called STUDENT. When logged onto the network, this directory is mapped as the DOS directory S:\WEB, as the DOS directory (mapped as root) W:\, as the NetWare directory STUDENT/KITHARA:WEB and as the Internet directory `http://www.su.edu/kithara`. Thus, when logged into the STUDENT NetWare file server as member of group KITHARA, the following all refer to the same directory.

```
S:\WEB
W:\
STUDENT/KITHARA:WEB
http://www.su.edu/kithara
```

It is a simple matter for the network/web administrator to map the Internet URL

```
http://www.su.edu/kithara
```

to the NetWare directory STUDENT/KITHARA:WEB. In this case, the Windows NT web server is attached to the NetWare server as user/process WEBMASTER and has "**Read**" and "**File Scan**" rights in that directory. Thus, instead of using FTP (file transfer program) to put files on the web server, a simple file copy operation can be used by both teacher and students.

IR019218

Each student is assigned a subdirectory in this web subdirectory with appropriate rights. Drive mappings provide each student convenient access to his or her own directory, while a web page in the root provides convenient access to any student web page for Internet browsers.

There are many ways to automate the required NetWare mappings. The following NetWare login script is used for Kithara network users.

```
IF FILE_SERVER = "STUDENT" THEN BEGIN
   IF MEMBER OF "KITHARA" THEN BEGIN
      WRITE "BEGIN KITHARA login script..."
      MAP S:=STUDENT/KITHARA:
      MAP ROOT U:=STUDENT/KITHARA:USER/%LOGIN_NAME
      MAP ROOT W:=STUDENT/KITHARA:WEB
      WRITE "END KITHARA login script."
      END
   END
```

Alternatively, if login scripts are not used, the student (or teacher) can create the drive mapping once and specify that the mapping is to be reconnected at login. (This is a feature of Windows 95).

However, creating and supporting the necessary infrastructure to make it easy for students (and teachers) to use this capability without having to know too much about the process is the topic of the rest of this paper.

ISN'T DOS DEAD?

To most users, DOS means the command line interface found in the program `command.com`. Let's face it. DOS was usable, but not easy or friendly. Since 1986, sophisticated users have used 4DOS in the form of `4dos.com` to replace `command.com` as their command-line processor. The advantage of 4DOS is that the command set and user interface is a large superset of DOS, providing most of the conveniences that have been provided in UNIX command line interfaces for several decades. 4DOS is very compatible with DOS, the only differences being in areas where the makers of 4DOS, JP Software, Inc., had to decide whether to duplicate certain bugs (or features) in DOS. For example, should 4DOS duplicate what the official documentation says `command.com` does or what `command.com` actually does.

4DOS is shareware that is available from JP Software, Inc. Additional information is available at their web site at http://www.jpsoft.com. And, since JP Software has similar versions available for OS/2, Windows NT, and Windows 95, the same batch files can be used with those systems.

But, why are we talking about DOS? Isn't DOS dead? Well, yes and no. Few users use DOS, and those that do use it to run legacy programs that still serve some useful purpose. But, some form of batch processing is still required in any system, unless one is willing to do every task, no matter how many times it is to be done, via point and click and keyboard input.

Programming languages can be used to automate such tasks. Visual Basic is the primary systems language in Windows. REXX is the primary systems language in OS/2 (and other IBM systems). Java is fast becoming a viable alternative for certain types of batch processing. Products such as

Novell GroupWise and IBM Lotus Notes/Domino have made their API (Applications Programmer Interface) available through Java applications (but not via applets, since there are additional security restrictions on applets that are not present for applications). But, for many batch tasks, 4DOS batch files fit the bill quite nicely, the other alternatives being just to complicated and difficult to develop and maintain for certain tasks.

BATCH FILES

A DOS batch file is a text file that contains DOS commands. A 4DOS batch file is a text file that contains 4DOS commands. The default file extension of a DOS batch file is `.bat`. The default file extension of a 4DOS batch file is `.btm`, which stands for batch file executed in memory. 4DOS loads the entire batch file into memory and executes it there rather than the DOS approach of loading each line from the file as it is needed, so that changes in the file during execution modify the behavior or the batch file. For example, the following batch file outputs the text "Hello, World.".

```
@ECHO OFF

ECHO Hello, World
```

The `@ECHO OFF` turns the ECHO off after this statement, and the `@` causes the ECHO not to be echoed. A 4DOS setting fixes this feature (or bug) so that every batch file does not have to begin with `@ECHO OFF`.

An ASCII text editor such as Notepad can be used to edit text files, whether they be data files or batch program files. The task can be simplified by creating a shortcut on the desktop with a command line that invokes Notepad and opens the desired file. For example, the following command invokes Notepad and opens the batch file `F:\USERS\hello.btm`.

```
C:\WINDOWS\notepad.exe F:\USERS\hello.btm
```

This command assumes that the Windows directory is `C:\WINDOWS`.

Running a batch file is simple. If the batch file were `F:\USERS\hello.btm`, the command to run the batch file would be `F:\USERS\hello.btm`. A shortcut to run the batch file would have the following command line.

```
G:\4DOS600\4dos.com /C F:\USERS\hello.btm
```

This command assumes that `4dos.com` is in the directory `G:\4DOS600`.

The redirection character ">!" can be used to redirect the output of a batch file to another file. For example, the command

```
F:\USERS\hello.btm >! F:\USERS\hello.out
```

causes the output of `hello.btm` to be sent to the file `hello.out`, overwriting any previous `hello.out` file.

We will now look at 4DOS batch files from the point of view of making it easy for students to create and maintain HTML web pages. The extensions to user and group management in a network environment are intimately related, as we shall see. Only those features of 4DOS that are necessary for this goal will be covered, leaving many other useful and powerful 4DOS features unmentioned.

ASSUMPTIONS

We have to start somewhere, so let's make the following assumptions.

A teacher (or network administrator) is going to create such an infrastructure to support students (or users) on a network which is connected to the Internet.

To avoid overly complicating the system, the removal of students from the system will not be addressed, but is left as an exercise for the reader. In practice, since it can be difficult to determine manually which directories should be deleted, the system will just grow larger until provisions are made to reduce the size of the directories.

The list of student names and userids for the students are in a text file, one per line, separated by the colon character ":". This text file could be created manually or created as the output of a database query or spreadsheet program. The text file `F:\USERS\users.dat` will be used for example purposes and appears as follows.

```
Snyder    : Tamara        : TAMARA
Snyder    : Gregory       : GREGORY
Snyder    : Cory          : CORY
Snyder    : Luke          : LUKE
Greenly   : Emily         : EMILY
```

For convenience, the user last name, first name, and userid are on the same line. This means that we will need to separate them in order to process the userid separately from the user name. The last name appears first so that the file can be easily sorted by last name.

The first batch file uses the environment variable G and the FOR command to ECHO every line of the text file `users.dat`.

```
FOR %G IN (@F:\USERS\users.dat) ECHO %G
```

The environment variable `F` is set to each of the lines of the text file and then echoed to the screen using the ECHO command.

But, in most cases, more than one statement is needed to accomplish the desired purpose. The previous program can be written as follows.

```
FOR %G IN (@F:\USERS\users.dat) GOSUB :PROCESS

QUIT

:PROCESS
```

```
   ECHO %G
   RETURN
```

We now need to separate the user name from the userid. The environment variable `U` will be used for the userid and the environment variable `N` will be used for the name. The following program does such a separation and outputs just the userid.

```
FOR %G IN (@F:\USERS\users.dat) GOSUB :PROCESS

QUIT

:PROCESS
   SET I=%@INDEX[%G,:]
   SET L=%@TRIM[%@SUBSTR[%G,0,%I]]
   SET F=%@TRIM[%@SUBSTR[%G,%@EVAL[%I+1],255]]
   SET I=%@INDEX[%F,:]
   SET U=%@TRIM[%@SUBSTR[%F,%@EVAL[%I+1],255]]
   SET F=%@TRIM[%@SUBSTR[%F,0,%I]]
   GOSUB :PROCESS1
   RETURN

:PROCESS1
   ECHO Last="%L"  First="%F"  UserID="%U"
   RETURN
```

The statements in `PROCESS` look imposing, but they are just used to split the line of text and set the environment variables `N` and `U`. Note that a feature of 4DOS is that since the comma "," is used to delimit arguments to `SUBSTR`, a comma cannot appear in the text file, as might be desired if the names were to be of the following form.

<last> , <first>

The output is as follows.

```
Last="Snyder"  First="Tamara"       UserID="TAMARA"
Last="Snyder"  First="Gregory"      UserID="GREGORY"
Last="Snyder"  First="Cory"         UserID="CORY"
Last="Snyder"  First="Luke"         UserID="LUKE"
Last="Greenly" First="Emily"        UserID="EMILY"
```

Note that we are not concerned with lining up the output. Note also the introduction of the procedure `PROCESS1`. To save space, unless otherwise specified, all program code will refer only to the body part of the `PROCESS1` procedure of the program. The `PROCESS1` statements of the above program are as follows.

```
ECHO Last="%L" First="%F"  UserID="%U"
```

The following `PROCESS1` statements create a user and web directory for each student, if that directory does not already exist.

```
IFF NOT ISDIR D:\WEB\%U THEN
      ECHO Making directory D:\WEB\%U
      *MD D:\WEB\%U
      ENDIFF

   IFF NOT ISDIR D:\USER\%U THEN
      ECHO Making directory D:\USER\%U
      *MD D:\USER\%U
      ENDIFF
```

But, just making the directory is not enough. Appropriate rights must also be given to the user in that directory. The following program grants the "**Read**", "**Write**", "**Create**", "**Erase**", "**Modify**", and "**File Scan**" rights to the user, all but the "**Access Control**" right.

The following `PROCESS1` statements grant the appropriate rights for each student.

```
ECHO.
   E:\PUBLIC\grant.exe R W C E M F FOR D:\WEB\%U TO USER %U
   E:\PUBLIC\grant.exe R W C E M F FOR D:\USER\%U TO USER %U
```

Note that many networks store the `grant.exe` command on the `Z:` drive.

This batch file is the same as issuing the following commands.

```
E:\PUBLIC\GRANT.EXE R W C E M F FOR D:\WEB\TAMARA TO USER TAMARA
E:\PUBLIC\GRANT.EXE R W C E M F FOR D:\USER\TAMARA TO USER TAMARA

E:\PUBLIC\GRANT.EXE R W C E M F FOR D:\WEB\GREGORY TO USER GREGORY
E:\PUBLIC\GRANT.EXE R W C E M F FOR D:\USER\GREGORY TO USER GREGORY

E:\PUBLIC\GRANT.EXE R W C E M F FOR D:\WEB\CORY TO USER CORY
E:\PUBLIC\GRANT.EXE R W C E M F FOR D:\USER\CORY TO USER CORY

E:\PUBLIC\GRANT.EXE R W C E M F FOR D:\WEB\LUKE TO USER LUKE
E:\PUBLIC\GRANT.EXE R W C E M F FOR D:\USER\LUKE TO USER LUKE

E:\PUBLIC\GRANT.EXE R W C E M F FOR D:\WEB\EMILY TO USER EMILY
E:\PUBLIC\GRANT.EXE R W C E M F FOR D:\USER\EMILY TO USER EMILY
```

Note that the `ECHO.` command is used to output a blank line.

Obviously, the batch method scales up to hundreds of users with very little additional effort. That is exactly the purpose of repetition and looping actions in programming.

Note that, in Novell NetWare, if a user were given the "**Access Control**" right, the user could inadvertently use that right to disallow themselves access to information in subdirectories of the directory to which they were granted rights. (Is this a feature or a bug?). In addition, in order to maintain individual responsibility for web directories, it is not advisable to allow users to permit

other users to place information in directories that appear on the Internet.

MASTER INDEX

The next step is to create a master index as the file `D:\index.htm`. This can be accomplished with the following program.

```
ECHO `<HTML>`
ECHO `<HEAD>`
ECHO `<TITLE>Kithara students</TITLE>`
ECHO `</HEAD>`
ECHO `<BODY>`
ECHO `<H1>Kithara students</H1>`
ECHO `<UL>`

FOR %G IN (@F:\USERS\users.dat) GOSUB :PROCESS

ECHO `</UL>`
ECHO `</BODY>`
ECHO `</HTML>`

QUIT

:PROCESS
   SET I=%@INDEX[%G,:]
   SET L=%@TRIM[%@SUBSTR[%G,0,%I]]
   SET F=%@TRIM[%@SUBSTR[%G,%@EVAL[%I+1],255]]
   SET I=%@INDEX[%F,:]
   SET U=%@TRIM[%@SUBSTR[%F,%@EVAL[%I+1],255]]
   SET F=%@TRIM[%@SUBSTR[%F,0,%I]]
   GOSUB :PROCESS1
   RETURN

:PROCESS1
   ECHO `<LI>`%L, %F - `<A` HREF="./%U/index.htm"`>`%U`</A>`
   RETURN
```

Note that since "<" and ">" are special characters (for redirection) in DOS, those characters need to be enclosed in backquotes (i.e., all but environment variables can be enclosed in backquotes).

The output is as follows.

```
<HTML>
<HEAD>
<TITLE>Kithara students</TITLE>
</HEAD>
<BODY>
<H1>Kithara students</H1>
<UL>
<LI>Snyder, Tamara - <A HREF="./TAMARA/index.htm">TAMARA</A>
<LI>Snyder, Gregory -<A HREF="./GREGORY/index.htm">GREGORY</A>
<LI>Snyder, Cory - <A HREF="./CORY/index.htm">CORY</A>
```

```
<LI>Snyder, Luke - <A HREF="./LUKE/index.htm">LUKE</A>
<LI>Greenly, Emily - <A HREF="./EMILY/index.htm">EMILY</A>
</UL>
</BODY>
</HTML>
```

Suppose that this program is called `F:\USERS\makelist.btm`. The following 4DOS command will run the batch file with the output redirected to the file `D:\WEB\index.htm`.

```
F:\USERS\makelist.btm >! D:\WEB\index.htm
```

Note that the command line from an icon shortcut would be as follows, where `4dos.com` is located in the directory `G:\4DOS600`.

```
G:\4DOS600\4dos.com /C F:\USERS\makelist.btm >! D:\WEB\index.htm
```

Now, anyone who can access the file `D:\WEB\index.htm` has convenient access to any of the student web pages.

Suppose that each student is supposed to create a web page as the file `index.htm` in their web directory (i.e., the directory created by the above program). Which students have web pages? The following `PROCESS1` statements display the userids of those students who do not have an `index.htm` file.

```
IFF NOT EXIST D:\WEB\%U\index.htm THEN
      ECHO User %U does not have a web page.
      ENDIFF
```

Note that we can assume that the directory does exist, because we just ran the previous program that updated the creation of the directories.

But, why not give users a default web page, if one does not already exist? The following `PROCESS1` statements creates a default web page for the student if and only if that student does not already have a web page. That is, we don't want to erase any of their previous work.

```
IFF NOT EXIST D:\WEB\%U\index.htm THEN
      ECHO Making web page for %U as D:\WEB\%U\index.htm.
      CALL F:\USERS\makepage.btm >! D:\WEB\%U\index.htm
      ENDIFF
```

The batch file `makepage.btm` can assume that the environment variables `U` and `N` have been set properly.

The batch file `makepage.btm` appears as follows.

```
ECHO `<HTML>`
ECHO `<HEAD>`
ECHO `<TITLE>`%F %L`</TITLE>`
ECHO `</HEAD>`
```

```
ECHO `<BODY>`
ECHO `<H1>`%F %L`</H1>`
ECHO `<UL>`
ECHO `<LI>`UserID: %U
ECHO `</UL>`
ECHO `<HR>`
ECHO `<A HREF=../index.htm>Back to student list</A>`
ECHO `</BODY>`
ECHO `</HTML>`
```

The default home page for user TAMARA appears as follows.

```
<HTML>
<HEAD>
<TITLE>Snyder  Tamara</TITLE>
</HEAD>
<BODY>
<H1>Snyder  Tamara</H1>
<UL>
<LI>UserID: TAMARA
</UL>
<HR>
<A HREF=../index.htm>Back to student list</A>
</BODY>
</HTML>
```

CONCLUSIONS

4DOS is an easy and powerful to create an infrastructure that makes it easy for students to create and maintain HTML web pages. This article has just touched on the use of 4DOS to automate such tasks.